ATM Switches

Switching Technology S38.3165
http://www.netlab.hut.fi/opetus/s383165

ATM switches

- General of ATM switching
- Structure of an ATM switch
- Example switch implementations
  - Knockout switch
  - Abacus
- Dimensioning example
General of ATM switching

- ATM switches correspond to layer 2 in the OSI reference model and this layer can roughly be divided into a higher and lower layer:
  - higher layer = ATM Adaptation Layer (AAL)
  - lower layer = ATM layer

ATM Adaptation Layer

AAL maps higher-layer information into ATM cells to be transported over an ATM network. At reception, AAL collects information from ATM cells for delivery to higher layers.

- AAL offers different service classes for user data
  - delay, bit rate and connection type (connectionless or circuit emulation) are the basic attributes of the service classes
- SAR (Segmentation and Reassembly) sub-layer for segmentation of variable length user data packets into fixed-size ATM cell payloads and at reception reassembly of ATM cell payload into user packets
- CS (Convergence Sub-layer) maps specific user data requirements onto ATM transport network
**ATM service classes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Class A</th>
<th>Class B</th>
<th>Class C</th>
<th>Class D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing relation between source and destination</td>
<td>Required</td>
<td>Not required</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bit rate</td>
<td>Constant</td>
<td>Variable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection mode</td>
<td>Connection-oriented</td>
<td>Connection-oriented</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALL (s)</td>
<td>AAL1</td>
<td>AAL2</td>
<td>AAL3/4 or AAL5</td>
<td>AAL3/4 or AAL5</td>
</tr>
<tr>
<td>Examples</td>
<td>E1, nx64 kbit/s emulation</td>
<td>Packet video, audio</td>
<td>Frame Relay X.25</td>
<td>IP, SMDS</td>
</tr>
</tbody>
</table>

**ATM Adaptation Layer 1 (AAL1)**

- **P format**
  - 1 octet SAR-PDU header
  - 1 octet Structure pointer
  - 46 octets User information

- **Non-P format**
  - SAR-PDU header
  - User information

**Notes:**
- SAR-PDU: Segmentation and Reassembly Packet Data Unit
- CS: Convergence Sub-layer
- CRC: Cyclic Redundancy Check
ATM Adaptation Layer 2 (AAL2)

CID - Connection Identification
HEC - Header Error Check
LI - Length Indicator
LLC - Logical Link Control
RES - Reserved

LLC packet header CID=1
User information
3 octets < 61 octets

LLC packet header CID=2
User information

ATM cell payload

OSF - Offset
SN - Sequence Number
STF - Start Field

ATM header STF

ATM header

ATM Adaptation Layer 3/4 (AAL3/4)

AAL3/4 CPCS PDU

4 octets < 65 535 octets

CS-PDU header
CS-PDU user information Pad CS-PDU trailer

AAL3/4 SAR PDU

2 octets

SAR-PDU header
SAR-PDU payload Pad SAR-PDU trailer

SAR-PDU header
SAR-PDU user information

SAR-PDU header
SAR-PDU user info length

CS - Convergence Sub-layer
CPCS - Common Part CS
CRC - Cyclic Redundancy Check
MID - Message Identifier
SAR - Segmentation and Reassembly
PDU - Packet data Unit
Btag - Beginning tag
BAsize - Buffer Allocation tag
Etag - Ending tag
ATM Adaptation Layer 5 (AAL5)

- Pad - Padding octets
- UU - AAL layer user-to-user indicator
- CPI - Common part indicator
- LEN - Length indicator
- CRC - Cyclic redundancy Check

ATM layer

ATM layer (common to all services) offers transport of data in fixed-size cells and also defines the use of virtual connections (VPs and VCs)

- multiplexing/demultiplexing of cells belonging to different virtual connections
- translations of inbound VPIs/VCLs to outbound VPIs/VCLs
- cell header generation for data received from AAL and cell header extraction when a cell is delivered to AAL
- flow control
General of ATM switching (cont.)

- ATM is a connection-oriented transport concept
  - an end-to-end connection (virtual channel) established prior to transfer of cells
  - signaling used for connection set up and release
  - data transferred in fixed 53 octets long cells (5 octets for header and 48 octets for payload)
- Cells routed based on two header fields
  - virtual path identifier (VPI) - 8 bits for UNI and 12 bits for NNI
  - virtual channel identifier (VCI) - 16 bits for UNI and NNI
  - combination of VPI and VCI determines a specific virtual connection between two end-points

ATM cell structure

```
<table>
<thead>
<tr>
<th>GFC</th>
<th>VPI</th>
<th>VCI</th>
<th>PTI</th>
<th>CLP</th>
</tr>
</thead>
</table>
| UNI - User Network Interface
| NNI - Network-to-Network Interface
| VPI - Virtual Path Identifier
| VCI - Virtual Channel Identifier
| GFC - Generic Flow Control
| PTI - Payload Type Identifier
| CLP - Cell Loss Priority
| HEC - Header Error Control

HEC = 8 x (header octets 1 to 4) / (x^4 + x^2 + x + 1)
```
General of ATM switching (cont.)

• VPI/VCI is determined on a per-link basis
  => VPI/VCI on an incoming link is replaced (at the ATM switch) with another VPI/VCI for an outgoing link
  => number of possible paths in an ATM network increased substantially (compared to having end-to-end VPI/VCIs)
• Each ATM switch includes a Routing Information Table (RIT), which is used in mapping incoming VPI/VCIs to outgoing VPI/VCIs
• RIT includes:
  • old VPI/VCI
  • new VPI/VCI
  • output port address
  • priority

General of ATM switching (cont.)

• When an ATM cell arrives to an ATM switch, VPI/VCI in the 5-octet cell header is used to point to a RIT location, which includes
  • new VPI/VCI to be added to an outgoing cell
  • output port address indicating to which port the cell should be routed
  • priority field allowing the switch to selectively send cells to output ports or discard them (in case of buffer overflow)
• Three routing modes:
  • unicast - \( \log_2 N \) bits needed to address a destination output port
  • multi-cast - \( N \) bits needed to address destined output ports
  • broadcast - \( N \) bits needed to address destined output ports
• In multi-cast/broadcast case, a cell is replicated into multiple copies and each copy is routed to its intended output port/outbound VC
General of ATM switching (cont.)

- ATM connections are either
  - pre-established - permanent virtual connections (PVCs)
  - dynamically set up - switched virtual connections (SVCs)
- Signaling (UNI or PNNI) messages carry call set up requests to ATM switches
- Each ATM switch includes a call processor, which
  - processes call requests and decides whether the requested connection can be established
  - updates RIT based on established and released call connections
    - ensuring that VPIs/VCIs of cells, which are coming from several inputs and directed to a common output are different
  - finds an appropriate routing path between source and destination ports

VPI/VCI translation along transport path

<table>
<thead>
<tr>
<th>Old VPI/VCI</th>
<th>New VPI/VCI</th>
<th>Output port</th>
<th>Priority field</th>
</tr>
</thead>
<tbody>
<tr>
<td>X P 15</td>
<td>Y Z 10</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>Y Z 10</td>
<td>Z W 8</td>
<td>P</td>
<td></td>
</tr>
</tbody>
</table>

RIT - Routing Information Table
VPI/VCI translation (cont.)

• VPI/VCI replacement usually takes place at the output ports
  => RIT split into two parts
  • input RIT - includes old VPI/VCI and N-bit output port address
  • output RIT - includes \( \log_2 N \)-bit input port address, old VPI/VCI
    and new VPI/VCI
• Since cells from different input ports can arrive to the same output
  port and have the same old VPI/VCI, the input port address is
  needed to identify uniquely different connections

ATM switches

• General of ATM switching
• **Structure of an ATM switch**
• Example switch implementations
  • Knockout switch
  • Abacus
• Dimensioning example
Main blocks

- **Line interface** cards (LICs), which implement input and output port controllers (IPC and OPCs)
- **Switch fabric** provides interconnections between input and output ports
- **Switch controller**, which includes
  - a call processor for RIT manipulations
  - control processor to perform operations, administration and maintenance (OAM) functions for switch fabric and LICs
Functions of input port controller

- Line termination and reception of incoming line signal
- Conversion of optical signal to electronic form if needed
- Decoding/descrambling of line/block coded or scrambled line signal
- Transport frame, e.g. SDH or PDH frame, processing
- Extraction of cell header for processing
- Storing of cell payload (or whole cells) to buffer memory
- HEC processing
  => discard corrupted cells
  => forward headers of uncorrupted cells to routing process
- Generation of a new cell header (if RIT only at input) and routing tag to be used inside the switch fabric
- Cell stream is slotted and a cell is forwarded through the switch fabric in a time-slot

Functions of output port controller

- Cells received from the switch fabric are stored into output buffer
- Generation of a new cell header (if RIT also at output)
- One cell at a time is transferred to the outgoing line interface
- If no buffering available then contention resolution
  => one cell transmitted and others discarded (or buffered at the inputs to wait for the next contention round)
- If buffering available and priorities supported then higher priority cells forwarded first to transport frame processing
- Cell encapsulation into transport frames, e.g. SDH or PDH frame
- Line/block encoding or scrambling of outgoing bit stream
- Conversion of electronic signal to optical form (if needed)
- Transmission of outgoing line signal
Input and output controller blocks

Input controller blocks:

Output controller blocks:

Switch control

- Switch controller implements functions of ATM management and control layer
- **Control plane**
  - responsible for establishment and release of connections, which are either pre-established (PVCs) using management functions or set up dynamically (SVCs) on demand using signaling, such as UNI and PNNI signaling
  - signaling/management used to update routing tables (RITs) in the switches
  - implements ILMI (Integrated Local Management Interface), UNI signaling and PNNI routing protocols
  - processes OAM cells
Switch control (cont.)

- **ILMI protocol** uses SNMP (Simple Network Management Protocol) to provide ATM network devices with status and configuration information related to PVCs, SVCs, registered ATM addresses and capabilities of ATM interfaces
- **UNI signaling** specifies the procedures to dynamically establish, maintain and clear ATM connections at UNI
- **PNNI protocol** provides the functions to establish and clear connections, manage network resources and allow network to be easily configurable
- **Management plane**
  - provides management functions and capabilities to exchange information between the user plane and control plane

ATM protocol reference model
Switch fabric

- Provides interconnections between input and output interfaces
- ATM specific requirements
  - switching of fixed length cells
  - no regular switching pattern between an input-output port pair, i.e., time gap between consecutive cells to be switched from an input to a specific output varies with time
- Early implementations used time switching principle (mostly based on shared media fabrics) - easy to use, but limited scalability
- Increased input rates forced to consider alternative solutions
  => small crossbar fabrics were developed
  => multi-stage constructions with self-routing reinvented

Cell routing through switch fabric

- Cells usually carried through switch fabric in fabric specific frames
- Carrier frames include, e.g. header, payload and trailer fields
- Header field sub-divided into
  - source port address
  - destination port address
  - flow control sub-field (single/multi-cast cell, copy indication, etc.)
- Payload field carries an ATM cell (with or without cell header)
- Trailer is usually optional and implements an error indication/correction sub-field, e.g. parity or CRC

General structure of a cell carrier frame

| Frame header | Frame payload | Frame trailer |
ATM switching and buffering

Due to asynchronous nature of ATM traffic, buffering is an important part of an ATM switch fabric design.

A number of different buffering strategies have been developed:

- input buffering
- output buffering
- input-output buffering
- internal buffering
- shared buffering
- cross-point buffering
- recirculation buffering
- multi-stage shared buffering
- virtual output queuing buffering

Buffering strategies

- Input buffered
- Output buffered
- Input-output buffered
- Internally buffered
- Shared buffer
Buffering strategies (cont.)

Cross-point buffered

Recirculation buffered

Multi-stage shared buffer

Virtual output queuing

ATM switching and buffering (cont.)

Input buffered switches

- Suffers from HOL blocking => throughput limited to 58.6 % of the maximum capacity of a switch (under uniform load)
- Windowing technique can be used to increase throughput, i.e. multiple cells in each input buffer are examined and considered for transmission to output ports (however only one cell transmitted in a time-slot)
  => window size of two gives throughput of 70 %
  => windowing increases implementation complexity
Output buffered switches

- No HOL blocking problem
- Theoretically 100 % throughput possible
- High memory speed requirement, which can be alleviated by concentrator
  => output port count reduced
  => reduced memory speed requirement
  => increased cell loss rate (CLR)
- Output buffered systems largely used in ATM switching

---

Input-output buffered switches

- Intended to combine advantages of input and output buffering
  - in input buffering, memory speed comparable to input line rate
  - in output buffering, each output accepts up to L cells \((1 \leq L \leq N)\)
  => if there are more than L cells destined for the same output, excess cells are stored in input buffers
- Desired throughput can be obtained by engineering the speed up factor \(L\), based on the input traffic distribution
- Output buffer memory needs to operate at \(L\) times the line rate
  => large-scale switches can be realized by applying input-output buffering
- Complicated arbitration mechanism required to determine, which \(L\) cells among the \(N\) possible HOL cells go to a contented output port
ATM switching and buffering (cont.)

Internally buffered switch

- Buffers implemented within switch blocks
- Example is a buffered banyan switch
- Buffers are used to store internally blocked cells
  \(\Rightarrow\) reduced cell loss rate
- Suffers from low throughput and high transfer delay
- Support of QoS requires scheduling and buffer management schemes
  \(\Rightarrow\) increased implementation cost

Shared-buffer switches

- All inputs and outputs have access to a common buffer memory
- All inputs store a cell and all outputs retrieve a cell in a time-slot
  \(\Rightarrow\) high memory access speed
- Works effectively like an output buffered switch
  \(\Rightarrow\) optimal delay and throughput performance
- For a given CLR shared-buffer switches need less memory than other buffering schemes
  \(\Rightarrow\) smaller memory size reduces cost when switching speed is high (\(\sim\) Gbits/s)
- Switch size is limited by the memory access speed (read/write time)
- Cells destined for congested outputs can occupy shared memory
  leaving less room for cells destined for other outputs (solved by assigning minimum and maximum buffer capacity for each output)
Cross-point buffered switches

- A crossbar switch with buffers at cross-points
- Buffers are used to avoid output blocking
- Each cross-point implements a buffer and an address filter
- Cells addressed to an output are accepted to a corresponding buffer
- Cells waiting in buffers on the same column are arbitrated to the output port one per time-slot
- No performance limitation as with input buffering
- Similar to output queuing, but the queue for each output is distributed over a number (N) of buffers => total memory space for a certain CLR > CLR for an output buffered system
- Including cross-point memory in a crossbar chip, limits the number of cross-points

Recirculation buffered switches

- Proposed to overcome output port contention problem
- Cells that have lost output contention are stored in circulation buffers and they contend again in the next time-slot
- Out-of-sequence errors avoided by assigning priority value to cells
- Priority level increased by one each time a cell loses contention => a cell with the highest priority is discarded if it loses contention
- Number of recirculation ports can be engineered to fulfill required cell loss rate (CLR = 10^{-6} at 80 % load and Poisson arrivals => recirculation port count divided by input port count = 2.5)
- Example implementations Starlite switch and Sunshine switch
  - Sunshine allows several cells to arrive to an output in a time-slot => dramatic reduction of recirculation ports
**ATM switching and buffering (cont.)**

**Multi-stage shared buffer switches**

- Shared buffer switches largely used in implementing small-scale switches - due to sufficiently high throughput, low delay and high memory utilization
- Large-scale switches can be realized by interconnecting multiple shared buffer switch modules
  => system performance degraded due to internal blocking
- In multi-stage switches, queue lengths may be different in the 1st and 2nd stage buffers and thus maintenance of cell sequence at the output module may be very complex and expensive

**Virtual output queuing switches**

- A technique to solve HOL blocking problem in input buffered switches
- Each input implements a logical buffer for each output (in a common buffer memory)
- HOL blocking reduced and throughput increased
- Fast and intelligent arbitration mechanism required, because all HOL cells need to be arbitrated in each time-slot
  => arbitration may become the system bottleneck
Design criteria for ATM switches

- Several criteria need to be considered when designing an ATM switch architecture
- A switch should provide bounded delay and small cell loss probability while achieving a maximum throughput (close to 100%)
- Capacity to support high-speed input lines (which possibly deploy different transport technologies, e.g. PDH or SDH)
- Self-routing and distributed control essential to implement large-scale switches
- Maintenance of correct cell sequence at outputs

Performance criteria for ATM switches

- Performance defined for different quality of service (QoS) classes
- Performance parameters:
  - cell loss ratio (CLR)
  - cell transfer delay (CTD)
  - two-point cell transfer delay variation (CDV)

Bellcore recommended performance requirements

<table>
<thead>
<tr>
<th>Performance parameter</th>
<th>CLP</th>
<th>QoS1</th>
<th>QoS3</th>
<th>QoS4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell loss ratio</td>
<td>0</td>
<td>&lt;10^{-10}</td>
<td>&lt;10^{-7}</td>
<td>&lt;10^{-7}</td>
</tr>
<tr>
<td>Cell loss ratio</td>
<td>1</td>
<td>N/S</td>
<td>N/S</td>
<td>N/S</td>
</tr>
<tr>
<td>Cell transfer delay (99th percentile)</td>
<td>1/0</td>
<td>150 μs</td>
<td>150 μs</td>
<td>150 μs</td>
</tr>
<tr>
<td>Cell delay variation (10^{-10} quantile)</td>
<td>1/0</td>
<td>250 μs</td>
<td>N/S</td>
<td>N/S</td>
</tr>
<tr>
<td>Cell delay variation (10^{-7} quantile)</td>
<td>1/0</td>
<td>N/S</td>
<td>250 μs</td>
<td>250 μs</td>
</tr>
</tbody>
</table>

N/S - not specified
Distribution of cell transfer delay

- Figure below shows a typical cell transfer delay distribution through a switch node.
- Fixed delay is attributed to table lookup delay and other cell header processing (e.g. HEC processing).
- For example:
  - Prob(CTD > 150 µs) < 1 - 0.99 => a = 0.01 and x = 150 µs (QoS1, 3 and 4)
  - Prob(CTD > 250 µs) < 10^{-10} => a = 10^{-10} and x = 250 µs (QoS1)

Cell processing times at different transmission speeds

- Processing time for different link speeds:
  - 221 µs/E1
  - 2.83 µs/STM-1
  - 708 ns/STM-4
  - 177 ns/STM-16

© P. Raatikainen  Switching Technology / 2007
Delay and jitter components

- Encapsulation/decapsulation delay
- Admission control (smoothing)
- Queuing delay
- Switching delay
- Transmission delay
- Propagation delay
- Reassembly (playtime) delay

No contribution to jitter
Contribution to jitter

ATM switches

- General of ATM switching
- Structure of an ATM switch
- Example switch implementations
  - Knockout switch
  - Abacus
- Dimensioning example
ATM switching fabric implementations

A lot of different switching network architectures have been experimented in ATM switch fabrics:

- Batcher-banyan based switches, e.g. Sunshine
- Clos network based switches, e.g. Atlanta
- Crossbar/crosspoint switches, e.g. TDXP (Tandem-Crosspoint)
- Ring and single/dual bus based switches

Most advanced ATM switching concepts are switching network independent, e.g. Knockout and Abacus

Knockout switch

- Output buffered switches largely used in ATM networks
- Capacity of output buffered switches limited by memory speed
- Problem solved by limiting the number of cells allowed to an output during each time-slot and excess cells discarded
  => knockout principle
- How many cells to deliver to an output port during each time-slot
  => this number can be determined for a given cell loss rate (CLR), e.g. 12 time-slots for CLR=10^-10, independent of switch size
- Memory speed seemed to be no more a bottleneck, however no commercial switch implementations appeared
  - inputs are supposed to be uncorrelated (not the case in real networks)
  - idea of discarding cells not an appealing one
- Knockout principle has been basis for various switch architectures
Knockout principle

• **N** input lines each implement a broadcast input bus, which is connected to every output block

• An output block is composed of cell filters that are connected to an **N-to-L** concentrator, which is further connected to a shared buffer

• No congestion between inputs and output blocks (NxN switch)

• Congestion occurs at the interfaces of outputs (inside concentrator)

• **k** cells passing through cell filters enter the concentrator and
  - if **k≤L** then all cells go to shared buffer
  - if **k>L** then L cells go to shared buffer and **k-L** cells are discarded

• Shared buffer includes a barrel shifter and **L** output (FIFO) buffers
  - barrel shifter stores cells coming from concentrator to FIFO memories in round robin fashion
  => complete sharing of output FIFO buffers

Knockout switch interconnection architecture

[Diagram of Knockout switch interconnection architecture]

- **Inputs**
- **Broadcast buses**
- **Outputs**
- **Bus interfaces**
Knockout switch bus interface

Operation of barrel shifter
Example construction of concentrator

Cell loss probability

- In every time-slot there is a probability $\rho$ that a cell arrives at an input.
- Every cell is equally likely destined for any output.
- $P_k$ denotes probability of $k$ cells arriving in a time-slot to the same output, which is a binomial distribution:
  \[ P_k = \binom{N}{k} \left( \frac{\rho}{N} \right)^k \left( 1 - \frac{\rho}{N} \right)^{N-k}, \quad k = 0, 1, \ldots, N \]
- Probability of a cell being dropped in N-to-L concentrator is given by:
  \[ P(\text{cell loss}) = \frac{1}{\rho} \sum_{k=L+1}^{N} (k-L) \binom{N}{k} \left( \frac{\rho}{N} \right)^k \left( 1 - \frac{\rho}{N} \right)^{N-k} \]
- Taking the limit as $N \to \infty$ and with some manipulation:
  \[ P(\text{cell loss}) = \left( 1 - \frac{L}{\rho} \right) \left( 1 - \sum_{k=0}^{\infty} \frac{\rho^k e^{-\rho}}{k!} \right) + \frac{\rho^L e^{-\rho}}{L!} \]
Cell loss probability (cont.)

Cell loss probability for some switch sizes (90% load)

- $N = 16$
- $N = 32$
- $N = 64$
- $N = \infty$

Cell loss probability for some load values ($N = \infty$)

- Load = 100%
- Load = 90%
- Load = 80%
- Load = 70%
- Load = 60%

Channel grouping

Channel grouping principle used in modular two-stage networks

- A group of outputs treated identically in the first stage
- A cell destined for an output of a group is routed to any output (at the first stage), which is connected to that group at the second stage
- First stage switch routes cells to proper output groups and second stage switches route cells to destined output ports
Channel grouping (cont.)

Asymmetric switch with line extension ratio of $KM/N$

- Output group of $M$ output ports corresponds to a single output address for the 1st stage switch
- At any given time-slot, $M$ cells at most can be cleared from a particular output group (one cell on each output port)

Maximum throughput per input

- increases with $K/N$ for a given $M$ (because load per output group decreases)
- increases with $M$ for given $K/N$ (because each output group has more ports for clearing cells)

<table>
<thead>
<tr>
<th>$M$</th>
<th>$K/N$</th>
<th>1/16</th>
<th>1/8</th>
<th>1/4</th>
<th>1/2</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0.061</td>
<td>0.117</td>
<td>0.219</td>
<td>0.382</td>
<td>0.586</td>
<td>0.764</td>
<td>0.877</td>
<td>0.938</td>
<td>0.969</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.121</td>
<td>0.233</td>
<td>0.426</td>
<td>0.686</td>
<td>0.885</td>
<td>0.966</td>
<td>0.991</td>
<td>0.998</td>
<td>0.999</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.241</td>
<td>0.457</td>
<td>0.768</td>
<td>0.959</td>
<td>0.996</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>0.476</td>
<td>0.831</td>
<td>0.991</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>0.878</td>
<td>0.959</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Channel grouping (cont.)

- Maximum throughput per input increases with M for given KM/N
- Channel grouping has a strong effect on throughput for smaller KM/N than for larger ones

<table>
<thead>
<tr>
<th>M</th>
<th>KM/N = 1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.586</td>
<td>0.764</td>
<td>0.877</td>
<td>0.938</td>
<td>0.969</td>
<td>0.984</td>
</tr>
<tr>
<td>2</td>
<td>0.686</td>
<td>0.885</td>
<td>0.966</td>
<td>0.991</td>
<td>0.998</td>
<td>0.999</td>
</tr>
<tr>
<td>4</td>
<td>0.768</td>
<td>0.959</td>
<td>0.966</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0.831</td>
<td>0.991</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.878</td>
<td>0.999</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>0.912</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>0.937</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>0.955</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>0.968</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>0.978</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>0.984</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Maximum throughput as a function of line expansion ratio KM/N

Multicast output buffered ATM switch (MOBAS)

- Channel grouping extends to the general Knockout principle
- MOBAS adopts the general Knockout principle
- MOBAS consists of
  - input port controllers (IPCs)
  - multi-cast grouping networks (MGN1 and MGN2)
  - multi-cast translation tables (MTTs)
  - output port controllers (OPCs)
MOBAS switch performance

- IPCs terminate incoming cells, look up necessary information in translation tables and attach information in front of cells so that the cells can properly be routed in MGNs
- MGNs replicate multi-cast cells based on their multi-cast patterns and send one copy to each addressed output group
- MTTs facilitate the multi-cast cell routing MGN2
- OPCs store temporarily multiple arriving cells (destined for their output ports) in an output buffer, generate multiple copies for multi-cast cells with a cell duplicator (CD), assign a new VCI obtained from a translation table to each copy, convert internal cell format to standard ATM cell format and finally send the cell to the next switching node
- CD reduces output buffer size by storing only one copy of a multi-cast cell - each copy is updated with a new VCI upon transmission

MOBAS architecture

IPC - Input Port Controller
MGN - Multi-cast Grouping Network
MTT - Multi-cast Translation Table
OPC - Output Port Controller
SM - Switching Module
SSM - Small Switch Module
CD - Cell Duplicator
Abacus switch

• Knockout switches suffer from cell loss due to concentration/channel grouping (i.e. lack of routing links inside switch fabric)
• In order to reduce CLR, excess cells are stored in input buffers => result is an input-output buffered switch
• Abacus switch is an example of such a switch
  • basic structure similar to MOBAS, but it does not discard cells in switch fabric
  • switching elements resolve contention for routing links based on priority level of cells
  • input ports store temporarily cells that have lost contention
  • extra feedback lines and logic added to input ports
  • distributed arbitration scheme allows switch to grow to a large size
ATM switches

- General of ATM switching
- Structure of an ATM switch
- Example switch implementations
  - Knockout switch
  - Abacus
- Dimensioning example

Dimensioning example

- An ATM-switch is to be designed to support 20 STM-4 interfaces. RIT will be implemented at the input interfaces. How fast should RIT lookup process be?
- Cells are encapsulated into frames for delivery through the switch fabric. A frame includes a 53-octet payload field and 3 octets of overhead for routing and control inside the switch fabric. What is the required throughput of the switch fabric?

Solution

- ATM cells are encapsulated into VC-4 containers, which include 9 octets of overhead and 9x260 octets of payload. One VC-4 container is carried in one STM-1 frame and each STM-1 frame contains 9x261 octets of payload and 9x9 octets of overhead. (See figure on the next slide)
Dimensioning example (cont.)

Solution (cont.)

- STM-4 frame carries 4 STM-1 frames and thus there will be
  \[4 \times 9 \times 260 / 53 = 176.6\] cells arriving in one STM-4 frame
- One STM-4 frame is transported in 125 \(\mu\)s
  \[=> 176.6 / 125 \mu s = 1412830.2\] cells will arrive to an input in 1 sec
  \[=>\] one RIT lookup should last no more than 707.8 ns
- Total throughput of the switch fabric is \[20 \times 1412830.2\] cells/s
- Since each cell is carried through the switch fabric in a container of
  56 octets, the total load introduced by the inputs to the switch fabric
  is \[20 \times 1412830.2 \times 56\] octets/s \[\approx 1.582 \times 10^9\] octets/s \[\approx 12,7\] Gbits/s